

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

**Навчально-методичний комплекс  
«Інститут післядипломної освіти» НТУУ «КПІ»**

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

Віталія Романкевича

(підпис)

“ ” 2020 р.

**Дипломний проект  
на здобуття ступіня «бакалавр»**

Зі спеціальності 123 «Комп'ютерна інженерія»

на тему: **Веб-орієнтована система обробки замовлень зоомагазину. База даних.**

Виконав: слухач групи ЗКІ-зп71

Татошвілі Іраклій Апрієвич

(підпис)

Керівник: д.т.н., доц. Віталій Романкевич

(підпис)

Консультант з нормоконтролю: к.т.н. доц. Ярослав Клятченко

(підпис)

Рецензент:

(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Слухач \_\_\_\_\_  
(підпис)

Київ – 2020 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет України  
«Київський політехнічний інститут»**

Навчально-методичний комплекс «Інститут післядипломної освіти» НТУУ «КПІ»

(повна назва)

Кафедра системного програмування і спеціалізованих комп'ютерних систем

(повна назва)

Ступінь \_\_\_\_\_ бакалавр

(назва ОКР)

Спеціальності \_\_\_\_\_ 123 «Компютерна інженерія»

(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій Романкевич

(підпис)

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**

**на дипломний проект слухача**

Татошвілі Іраклія Апрієвича

1. Тема проекту «Веб-орієнтована система обробки замовлень зоомагазину. База даних» і керівник проекту д.т.н., доц. Віталій Романкевич, затверджені наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Строк подання студентом проекту \_\_\_\_\_

3. Вихідні дані до проекту див. Технічне завдання.

4. Зміст розрахунково-пояснювальної записки:

– виконати огляд існуючих рішень серед Веб-орієнтованих систем обробки замовлень зоомагазину та обґрунтування теми дипломного проекту;

– опис інструментарію;

– опис модуля бази даних Веб-орієнтованої системи обробки замовлень зоомагазину;

– опис та реалізація алгоритмів

5. Перелік графічного матеріалу:

–схема структурна інформаційної системи;

—схема бази даних модуля Веб-орієнтованої система обробки замовлень зоомагазину;

—схема алгоритму імпортування даних в базу даних;

—схема алгоритму реєстрації користувачів в базі даних Веб-орієнтованої система обробки замовлень зоомагазину.

6. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	к.т.н. доц. Ярослав Клятченко		

7. Дата видачі завдання \_\_\_\_\_

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1	Вивчення літератури за тематикою роботи	18.03.2020	
2	Складання та узгодження технічного завдання	21.03.2020	
3	Огляд існуючих рішень	25.03.2020	
4	Аналіз предметної області Web-орієнтованих система обробки замовлень зоомагазину.	28.03.2020	
5	Аналіз можливості імпортування даних про товари та послуги з CSV-файлу	13.04.2020	
6	Створення та програмна реалізація алгоритмів	13.04.2020	
7	Тестування алгоритмів	25.04.2020	
8	Підготовка графічної частини дипломного проекту	12.05.2020	
9	Оформлення документації дипломного проекту	16.05.2020	
10	Попередній огляд матеріалів диплому на кафедрі	20.05.2020	

Слухач

\_\_\_\_\_  
(підпис)

Іраклій Татошвілі

Керівник проекту

\_\_\_\_\_  
(підпис)

Віталій Романкевич

## АНОТАЦІЯ

Дипломний проект виконаний на здобуття ступіня “Бакалавр” з напрямку підготовки 123 “Комп’ютерна інженерія” та включає пояснювальну записку 64 стр., 4 табл., 18 рис.

Об’єкт розробки - модуль управління базою даних Веб-орієнтованої система обробки замовлень зоомагазину.

Ціль розробки – створення модуля керування базою даних, який забезпечує роботу з користувачами та товарами наявними в Веб-орієнтованій системі обробки замовлень зоомагазину.

Модуль дозволяє реєструвати користувачів, товари в базі даних, здійснювати пошук серед наявних у базі даних інформаційних блоків, виходячи з ресурсних потреб та з огляду на кількість користувацьких запитів, а також надає можливість імпортування даних до бази даних з зовнішніх ресурсів, таких як файли та інформаційні таблиці.

У роботі проаналізовано діючий прототип Веб-орієнтованої системи обробки замовлень зоомагазину, а саме модуль бази даних, реалізовано роботу з імпортуванням даних з інформаційних таблиць, файлів та інших Веб-ресурсів, досліджено різні підходи до реєстрації користувачів, запропоновано та реалізовано алгоритми імпортування даних, створено засоби для організації роботи з розподіленими базами даних та реалізується механізм управління даними з зручного Веб-інтерфейсу .

У процесі розробки були використані мова програмування Java, JavaScript, HTML, XML серверна технологія Apache TomCat 7.0 та СУБД MySQL Server 2015.

Ключові слова: база даних, Веб-орієнтована система, обробка замовлень зоомагазину.

## **ANNOTATION**

This diploma project is completed for the obtaining “Bachelor” degree in 123 “Computer Engineering” and consists of 64 pages of explanatory note, 4 tables and 18 pictures.

Subject of the development – database management module for Web-based order processing system of the zoo-store.

Development goal – creation of database management module which provides users with the ability to manage products and services of the Web-based order processing system of the zoo-store.

Module allows to register users, products in the database, to search through the available database information blocks, according to the resource needs and numbers of the user’s requests, it also allows importing data into the database from external resources such as files and information tables.

The document analyzes the working prototype of the Web-based order processing system of the zoo-store, its database module, to be precise. As a result of this work were developed and implemented such processes as importing data from information tables, files and other Web-resources, were explored different approaches to user registration, were proposed and implemented search algorithms created for the management of the distributed databases, and convenient data management mechanisms were also successfully implemented.

The technologies that were used in the process of the developing are the following: Java, JavaScript, HTML, XML server technology and Apache TomCat 7.0 database MySql Server 2015.

Keywords: database management, Web-based system, order processing system of the zoo-store.

[illegible]

[illegible]

## ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1 Вимоги до програмного продукту, що розробляється	3
5.2 Вимоги до апаратного та програмного забезпечення	3
6. ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.045490.002 ТЗ			
<b>Зм</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>	Веб-орієнтована система обробки замовлень зоомагазину Модуль управління. Технічне завдання	<b>Лит.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<b>Розроб.</b>		Татошвілі І.А.					1	4
<b>Перев.</b>		Романкевич В.О.						
<b>Н. контр.</b>		Клятченко Я.М.				НТУУ «КПІ імені Ігоря Сікорського» , ІПО, ЗКІ-зп71		
<b>Затв.</b>		Романкевич В.О.						



## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання використовується для розробки модуля бази даних Веб-орієнтованої системи обробки замовлень зоомагазину. Область використання: для використання в Веб-орієнтованих системах обробки замовлень зоомагазину у роботі з реєстрацією користувачів, створенні замовлень та імпортуванні та менеджменті інформаційних ресурсів(таблиць, файлів, зображень) з подальшим їх зберіганням у базі даних.

## 2.ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставами для розробки є завдання на виконання дипломного проекту ступіня “бакалавр” зі спеціальності 123 “Комп’ютерна інженерія”, затверджене кафедрою спеціалізованих комп’ютерних систем Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського”.

## 3.МЕТА РОЗРОБКИ

Метою даного проекту є спрощення уніфікація та підвищення ефективності роботи Веб-орієнтованої системи обробки замовлень зоомагазину за рахунок застосування нових підходів до проектування, покращених алгоритмів імпортування даних та нових технологій програмування в модулі бази даних вищезазначеної Веб-орієнтованої системи.

## 4.ДЖЕРЕЛА РОЗРОБКИ

Джерелами розробки є науково-технічна література, потреби користувачів та адміністраторів Веб-орієнтованих системи обробки замовлень зоомагазину, опубліковані зарубіжні та вітчизняні аналоги.

					ІАЛЦ.045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

## 5.ТЕХНІЧНІ ВИМОГИ

### 5.1.Вимоги до програмного продукту, що розробляється

Програма повинна забезпечувати такі основні функції:

- Додавання, редагування, копіювання та видалення користувачів та їх персональних даних
- Додавання, редагування, копіювання та видалення замовлень
- Імпортування та подальше розміщення в базі даних товарів, списків користувачів, з інформаційних CSV таблиць, файлів
- Імпортування та подальше розміщення в базі даних графічної та текстової інформації, з файлів, Web-ресурсів
- У випадку імпортування даних про товари та послуги з CSV-файлу, передбачити можливість автоматичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів

### 5.2.Вимоги до апаратного та програмного забезпечення

Вимоги до програмного забезпечення:

- операційна система Microsoft Windows (XP, Vista, 7, 8, 10) Linux, MacOS
- Apache TomCat 7.0 server, Java SE Development Kit 7, MySQL Database, MySQL Connectors (JDBC), MySQL Workbench

Вимоги до апаратного забезпечення:

- мінімальні апаратні вимоги для роботи вищезазначених програмних продуктів та операційних систем

					ІАЛІЦ.045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

## 6. Етапи розробки

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1	Вивчення літератури за тематикою роботи	18.03.2020	
2	Складання та узгодження технічного завдання	21.03.2020	
3	Огляд існуючих рішень	25.03.2020	
4	Аналіз предметної області Web-орієнтованих система обробки замовлень зоомагазину.	28.03.2020	
5	Аналіз можливості імпортування даних про товари та послуги з CSV-файлу	13.04.2020	
6	Створення та програмна реалізація алгоритмів	13.04.2020	
7	Тестування алгоритмів	25.04.2020	
8	Підготовка графічної частини дипломного проекту	12.05.2020	
9	Оформлення документації дипломного проекту	16.05.2020	
10	Попередній огляд матеріалів диплому на кафедрі	20.05.2020	

					ІАЛІЦ.045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

[illegible]

[illegible]

## Зміст

Список скорочень та позначень.....	3
Вступ.....	4
1. Аналіз існуючих рішень та обґрунтування теми дипломного проекту.....	5
1.1 Поняття Web-орієнтована система обробки замовлень та особливості її функціонування.....	7
1.2 Огляд існуючих зарубіжних Веб-орієнтованих систем обробки замовлень.....	10
1.3 Обґрунтування потреби у новій вітчизняній Web-орієнтованій системі обробки замовлень націленої на іноземного споживача.....	17
2. Опис інструментарію.....	17
2.1 Операційна система.....	17
2.2 Середовище розробки, платформа та мова програмування.....	17
2.3 Система управління базами даних.....	29
3. Опис модуля бази даних Веб-орієнтованої системи обробки замовлень зоомагазину.....	34
3.1 Загальна інформація про модулі Веб-орієнтованої системи обробки замовлень зоомагазину .....	34
3.2 Проектування модуля бази даних.....	37
3.2.1 Моделі даних і схеми в якості засобу збору даних.....	39
4. Опис алгоритмів.....	56
4.1 Покрокова реалізація алгоритму.....	57
4.2 Забезпечення імпортування даних з CSV-файлу, з можливістю автоматичного отримання зображень з Веб-ресурсу, та подальшого їх форматування до заданих розмірів.....	59
<b>ВИСНОВКИ</b>	
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ.....	63

						ІАЛЦ.045490.004 ПЗ		
<i>Зм</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>				
<i>Розроб.</i>		Татошвілі І.А.			Веб-орієнтована система обробки замовлень зоомагазину База даних. Пояснювальна записка	<i>Літ.</i>	<i>Аркуш</i>	<i>Архивів</i>
<i>Перев.</i>		Романкевич В.О.					1	54
<i>Н. контр.</i>		Клятчєнко Я.М.				НТУУ «КПІ імені Ігоря Сікорського», ІПО, ЗКІ-зп71		
<i>Затв.</i>		Романкевич В.О.						

Додатки

ДОДАТОК 1. Копії графічних матеріалів

ІАЛЦ.045490.005 Д1 Веб-Орієнтована система. Схема структурна

ІАЛЦ.045490.006 Д2 Модель бази даних. Схема структурна

ІАЛЦ.045490.007 Д3 Пошук пріоритетів. Схема алгоритму

ІАЛЦ.045490.008 Д4 Створення замовлень. Схема алгоритму

ДОДАТОК 2. Лістинг програми.

ДОДАТОК 3. Довідка про впровадження.

					ІАЛЦ.045490.004 ІЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підп.	Дата		

## СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

API – application programming language

CSS – cascading style sheets

HTTP – hyper text transfer protocol

IDE – integrated drive electronics

JDK – java development kit

JDBC – java data base connectivity

JMS – java message service

JMX – java management extension

JPS – java pages server

JRE – java runtime environment

JVM – java virtual machine

MS – Microsoft

MVC – model view container

ORM – object-relation mapping

SQL – structured query language

STS – spring tool suite

ОС – операційна система

ПК – персональний комп'ютер

СУБД – система управління базами даних

СДТП – система дистрибуції туристичних послуг

					ІАЛІЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3



## ВСТУП

Роль інформації в сучасному світі тяжко перебільшити, саме вона являється керівною силою будь-якої організації. Які поточні операції виконуються? Як провести оцінку діяльності та розробити стратегію? Які операції можна виконувати в майбутньому? Організаційна інформація зберігається в базі даних. Програми та програмні модулі, такі як системи управління дистрибуцією, постачанням, системи управління взаємовідносинами з клієнтами, реалізуються через доступ до інформації в базі даних, шляхом запитів програми і отримання відповідних даних зі сховища.

В моделі реляційної бази даних можуть зберігатися будь-які об'єкти чи сутності незалежно від того чи то особа, місце, річ, угода, або подія, все це, може зберігатися в базі даних. Технологія баз даних так важлива в бізнес-моделі Web-орієнтованих систем дистрибуції, бо в ієрархічній моделі баз даних інформація організована у вигляді деревовидної структури, що дозволяє ефективно використовувати її, покладаючись на батьківські чи дочірні відносини, таким чином, що кількість зв'язків зводиться до мінімуму.

Мережева модель бази даних має гнучкий спосіб представлення об'єктів та їх відносин, де структурна модель ієрархічних даних подана у вигляді дерева записів що має один батьківський запис і багато дітей. Така модель мережі дозволяє кожному запису мати кілька батьківських і дочірніх записів, утворюючи ґратчасту структуру. А саме мова йде про реляційну модель бази даних, що зберігає інформацію у вигляді логічно-пов'язаних двовимірних таблиць. Ця робота присвячена саме такому модулю бази даних Web-орієнтованої системи дистрибуції туристичних послуг оснований на реляційній базі даних.

					ІАЛЩ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

## 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

Web-орієнтовані системи обробки замовлень на даний момент не являються чимось принципово новим чи інноваційним, в той же час з ростом користувацького навантаження на мережі та бази даних постає зовсім інше питання пов'язане з організацією підвищення ефективності функціонування системи в цілому. Незалежно від того чи це мале, чи середнє підприємство, розв'язання проблеми ефективності модуля бази даних лежить не в площині кваліфікації персоналу, а саме в гнучкому використанні інноваційних технологій, та досконалому розумінні як логічних зв'язків так і технологічних можливостей на базі яких ці моделі можуть бути реалізовані.

Доцільним буде розглянути іноземних аналогів та приватного сектору, бо саме тут ми можемо знайти діючі і відносно ефективні моделі побудови інформаційних систем основаних на реляційній базі даних. Так чи інакше а Web-орієнтовані системи обробки замовлень являються підвидом електронної комерції, тому спочатку розглянемо її в цілому.

**Електронна комерція (e-commerce)** – це організація більшості бізнесів–процесів підприємства за рахунок їхнього проведення електронним чином (Internet, мереж стільникового зв'язку, внутрішніх локальних мереж підприємств). Сюди відносять електронний обмін інформацією (Electronic Data Interchange, EDI), електронний рух капіталу (Electronic Funds Transfer, EDF), електронну торгівлю (E-trade), обіг електронних грошей (E-cash), електронний маркетинг (E-marketing), електронний банкінг (E-banking), електронні страхові послуги (E-Insurance) тощо.

Кожен з видів електронної комерції включає наступні операції:

- встановлення контактів між потенційними контрагентами;
- обмін інформацією;
- купівлю–продаж товарів та послуг;

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

- до- і після продажну підтримку (надання інформації про товар або послугу, рекомендації щодо використання товару, консультації замовників);
- електронні платежі;
- управління доставкою;
- бізнес-адміністрування.
- Електронна комерція ґрунтується на використанні різних технологій, зокрема таких як: EDI (Electronic Data Interchange – електронний обмін даними), електронна пошта, Інтернет, інтранет (обмін інформацією всередині компанії), екстранет (обмін інформацією з зовнішнім світом).

Таким чином, електронну комерцію можна характеризувати як ведення бізнесу через Інтернет.

#### **Переваги та характерні риси електронної комерції.**

Електронна комерція має багато переваг:

- значно збільшується оперативність одержання інформації, особливо при міжнародних операціях;
- значно скорочується цикл виробництва і продажу, тому що немає необхідності щораз вводити отримані документи,
- знижується імовірність виникнення помилок під час введення документів;
- значно знижуються витрати, пов'язані з обміном інформацією за рахунок використання більш дешевих засобів комунікацій;
- використання інтернет-технологій електронної комерції дозволяє компанії стати більш відкритою по відношенню до клієнтів;
- використання інтернет-технологій електронної комерції дозволяє легко і швидко інформувати партнерів і клієнтів про продукти і послуги;

					ІАЛЦ.045490.004 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підп.	Дата		

- використання інтернет–технологій електронної комерції дозволяє створювати альтернативні канали продажів, наприклад, через електронний магазин на корпоративному сайті.

### **1.1 Поняття Web-орієнтована система обробки замовлень та особливості її функціонування**

Web-орієнтована система обробки замовлень є однією з найпоширеніших комерційних моделей електронної торгівлі у сфері B2C.

**Web-орієнтована система обробки замовлень** - являє собою спеціалізований Web–сайт, який належить фірмі–товаровиробнику, торговій фірмі тощо та призначений для просування споживчих товарів на ринку, збільшення обсягів продажу, залучення нових покупців.

Особливими рисами Web-орієнтованих систем обробки замовлень є те, що вони можуть забезпечувати споживачів значно більшим обсягом інформації, необхідної для прийняття рішення про покупку, та пропонувати значно більшу кількість товарів та послуг, ніж реальні магазини. Крім того використання Internet-технологій дозволяє здійснювати персональний підхід до споживачів з урахуванням попередніх відвідувань магазину та використовувати Web-орієнтовану систему для маркетингових досліджень (анкетування, конференції покупців і т.п.).

Превагою торгівлі через Web-орієнтовану системи є те, що вони потребують значно менших витрат на утримання та організацію роботи за рахунок обмеженої матеріально-технічної бази (будівель, споруд, приміщень та кількості обслуговуючого персоналу).

Проте Web-орієнтовані системи мають і недоліки. Основними є невизначеність реального існування товару та відповідність його основним параметрам якості, шахрайства при проведенні грошових транзакцій, проблеми з доставкою.

### **Класифікація Web-орієнтованих систем обробки замовлень**

Існує кілька систем класифікації Web-орієнтованих систем обробки замовлень:

					ІАЛЦ.045490.004 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підп.	Дата		

1. За методом роздрібного продажу товарів у мережі: Internet-магазини; Web-вітрини, торгові системи; торгові ряди; контентні проекти (споживацькі енциклопедії, системи Internet-замовлень товарів тощо);

2. За бізнес-моделлю: повністю онлайнвий магазин та суміщення офф-лайнного бізнесу з он-лайнним (коли Web-орієнтована система створюється на основі вже діючої реальної торгової структури);

3. За взаємовідносинами з постачальниками: магазини, які володіють власним складом (наявність реальних товарних запасів); магазини, що працюють за договорами з постачальниками (відсутність значних товарних запасів);

4. За ступенем автоматизації серед торгових систем електронних магазинів розрізняють Web-вітрини, власне Internet-магазини та торгові Internet-системи (TIS).

**Web-вітрина** – це спеціалізований Web-сайт, на якому пропонується товарний каталог, система навігації та оформлення замовлення з наступною передачею його менеджеру для оформлення. Той у свою чергу зв'язується зі складом, організовує доставку товару покупцеві, контролює процес оплати за товар. Паралельно ведеться рекламна робота, вивчення попиту, аналітична робота.

Робота Web-вітрини багато в чому нагадує посилкову торгівлю, проте замовлення на товар та ознайомлення з ним робиться через мережу Internet. В даному випадку Web-вітрина є інструментом залучення покупця, інтерфейсом для взаємодії з ним та проведення маркетингових заходів.

Web-орієнтована система обробки замовлень на відміну від Web-вітрини, характеризується повною автоматизацією, що дає можливість працювати індивідуально з кожним зареєстрованим клієнтом.

#### **Організація обслуговування покупців у віртуальних магазинах**

Типова схема взаємодії покупця з Web-орієнтованою системою здійснюється наступним чином :

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

1. Відкриття сайту, де представлено каталог товарів та необхідні елементи інтерфейсу для виконання операцій відбору та купівлі товарів;
2. Перегляд товарного каталогу та вибір товарів (формування кошика покупця);
3. Реєстрація покупця;
4. Вибір форми оплати та доставки товару;
5. Підтвердження замовлення;
6. Оплата товару;
7. Доставка придбаного товару покупцеві.

Пошук товарів в Web-орієнтованій системі може здійснюватися за допомогою каталогу або через внутрішню пошукову систему. За допомогою каталога можна здійснити огляд товару (найчастіше у вигляді фото), ознайомитися з його споживчими та технічними характеристиками (у вигляді тексту та спеціальних символів), ціною.

У процесі перегляду і відбору товару покупець формує власний віртуальний кошик, який являє собою список відібраного товару з вказанням його ціни, кількості та загальної вартості (з урахуванням можливих знижок). Коли зроблено остаточний вибір товару, покупцеві слід підтвердити замовлення, зареєструватися за встановленою процедурою з визначенням форми оплати та доставки товару.

Реєстрація полягає у заповненні покупцем спеціальної форми, що включає в себе інформацію про покупця, його пошту і (або)електронну адресу, особистий пароль та деяку іншу.

Під час проведення реєстрації особиста інформація покупця забезпечується шляхом передачі даних з використанням спеціальних методів захисту. Такими засобами можуть виступати протоколи SET або SSL (детальніша інформація про ці протоколи буде подана у розгляді теми 7).

Найпоширенішими формами оплати товарів, придбаними в Internet-магазині можна назвати:

1. Оплата кур'єрові готівкою здійснюється при доставці ним товару за вказаною адресою.

					ІАЛПЦ.045490.004 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підп.	Дата		

2. За допомогою платіжних систем мережі Internet.

3. Оплата поштовим (телеграфним) переказом.

4. Післяплата при одержанні товару на пошті.

5. Оплата банківським переказом.

Доставка придбаних товарів покупцю може здійснюватися такими способами:

1. Доставка власною чи професійною кур'єрською службою.

2. Доставка міжнародною кур'єрською службою.

3. Доставка поштою. Товар передається покупцю бандероллю з замовленням у відповідному поштовому відділенні.

4. Доставка магістральним (залізничним, автомобільним, водним, повітряним) транспортом. Такий вид доставки характерний для сегменту B2B при взаємодії виробників з посередниками та посередників між собою. Найчастіше так поставляються великогабаритні товари або великі партії товару. Ціни і терміни доставки залежать від конкретних умов угоди.

Перспективною формою торгівлі в системі B2C є електронні моли (E-mall) або електронні торговельні центри. Електронний мол являє собою Web-сайт, який містить значну кількість електронних крамниць та каталогів, об'єднаних загальним місцем розташування (в деяких випадках - під однією відомою маркою), що спільно виконують додаткові функції.

## 1.2 Огляд існуючих зарубіжних Web-орієнтованих систем обробки замовлень

Як вже зазначалося на початку розділу розгляд вітчизняних Web-орієнтованих систем обробки замовлень не є доцільним в силу їх слабкої розвиненості та відсутності тих актуальних для сьогоденності проблем, які дана робота поставила собі за ціль вирішити. Для ознайомлення з асортиментом наявних у інтернеті Web-орієнтованих систем що відповідають необхідним для аналізу критеріям було обрано одні з найкращих, а саме :

Американський - **HeavenlyPetStore**

					ІАЛЦ.045490.004 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підп.	Дата		

Британський - **Pets Corner**

Німецький – **Zoo Plus**

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		11



HeavenlyPetStore - це американський зоомагазин, який пропонує широкий асортимент товарів для домашніх улюбленців.

Welcome to our store

🛒

Cart

🔍

Search

HeavenlyPetStore

Home

Catalog

Pets

Pet Nest/Bed

Jewelry

Toys&Costumes

Contact Us

Save While Shopping!

Shop Now

Рис.1 HeavenlyPetStore

Зоомагазин має в розпорядженні наступні позиції:

- аксесуари

- ліжаки

- ювелірні вироби

- іграшки

- одяг

Зоомагазин виконую доставку у всі регіони всередині країни та за її межі, приймає до сплати на ступні методи:

- shopPay

- googlePay

					ІАЛІЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		12

- applePay
- payPal
- credit card

Магазин має інтуїтивний UI/UX. Дозволяю легко пройти реєстрацію для роботи с замовленням, його трекінгом та відгуком.

**Pets Corner** - це британський зоомагазин, який пропонує дуже широкий асортимент товарів та послуг для різноманітних тварин - ваших домашніх улюбленців.

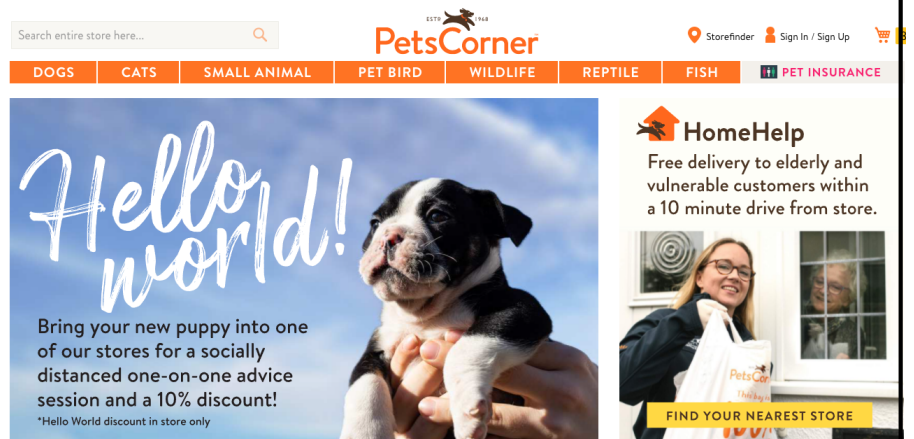


Рис.2 Pets Corner

Зоомагазин має в розпорядженні наступні позиції:

- аксесуари
- ліжаки
- ювелірні вироби
- іграшки
- одяг
- трекари
- присторою для харчування

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		13

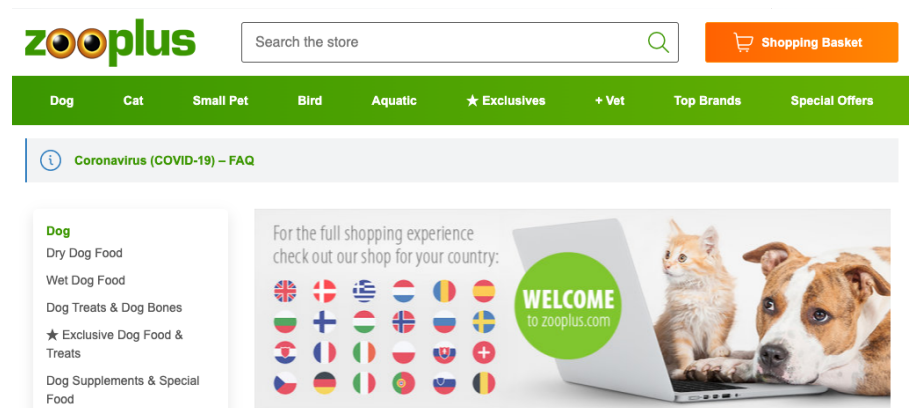
- надає послуги грумінгу
- надає послуги страхування

Зоомагазин виконує доставку у всі регіони всередині країни та за її межі, приймає до сплати наступні методи:

- PayPal
- credit card

Магазин є прикладом того, як має бути побудований сервіс, з дуже широким асортиментом, багатою інформаційною базою, засобами доставки, персональним кабінетом покупця. Можливістю обрати підписку на покупку товару з доставкою на кастомний період. Інтуїтивний UI/UX. Така модель системи обробки замовлень надає переваги для ведення бізнесу в електронній комерції в секторі.

**Zoo Plus** - це німецький зоомагазин, який пропонує повний асортимент товарів для домашніх тварин для вас та ваших домашніх тварин - понад 8000 найкращих продуктів на складі та готові до доставки. Маючи понад 20 років досвіду в галузі корму для домашніх тварин та аксесуарів для домашніх тварин, ви можете сподіватися на провідні бренди та якісні товари за надзвичайно низькими цінами.



					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

### Рис.3 Zoo Plus

Мета - полегшити життя за допомогою простого замовлення в Інтернеті та зручної доставки додому. Завдяки корисним функціям, таким як відео про продукцію, відгуки клієнтів та фотографії та ретельно досліджені описи товарів, ви завжди можете прийняти обгрунтоване рішення про те, який продукт підходить для вихованця.

У європейських офісах у Мюнхені, Страсбурзі, Кракові, Генуї та Оксфорді члени команди з понад 15 національностей працюють разом, щоб покращувати зооплус щодня. Магазин з гордістю надає все найкраще для вихованців 365 днів у році.

Важлива особливість в тому, що магазин ніколи не ділиться особистими даними з рекламодавцями чи іншими сторонніми особами. Дані використовуються лише для одного: обробка замовлення.

Магазин використовує 128-бітове шифрування SSL для всіх транзакцій.

Зоомагазин має в розпорядженні наступні позиції:

- аксесуари
- ліжаки
- ювелірні вироби
- іграшки
- одяг
- присторою для харчування

Та багато іншого, в тому числі сервіси по догляду за улюбленцем, для відпусток.

Важливою відмінністю від вище описаних магазинів є наявність ветеринарної продукції.

					ІАЛІЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		15

Магазин має дуже потужну інформаційну базу, можливість оформити підписку на новини. Інтуїтивний UI/UX.

### **Важливі аспекти технологічної реалізації модуля баз даних в Web-орієнтованих системах обробки замовлень зоомагазину**

Дослідивши вищезазначені Web-орієнтовані системи з обробки замовлень, а також користуючись можливістю дізнатися на які технології опиралися розробники успішних та ефективних Web-орієнтованих систем в цілому ми прийшли до розуміння які саме технології мають бути використані при створенні аналогічного як мінімум рівного по ефективності, продукту, за умови використання лише програмного забезпечення що знаходиться у вільному доступі, з метою мінімізації витрат на побудову функціональної моделі, а також з метою побудови діючого бізнес рішення для вітчизняної галузі, без залучення додаткових коштів.

### **1.3 Обґрунтування потреби у новій вітчизняній Web-орієнтованій системі обробки замовлень націлених на іноземного споживача.**

Як вже зазначалося існуюче на ринку України рішення не дозволяє а ні сприяти вітчизняному розвитку цього сектору електронної комерції, а ні забезпечує створення загальної Web-площини для реалізації продуктів зоологічної направленості. Основні напрямки, реалізації яких не вистачає для більш плідної співпраці держави з вищезгаданої індустрією це:

- дослідження та впровадження нових ідей в індустрії електронної комерції в зоосекторі
- поради про те, як почати і розвивати цей бізнес
- створення спеціальної податкової пільги
- останні новини індустрії зоомакету
- схеми забезпечення якості

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		16

- програми розвитку менеджменту
- онлайн і офлайн маркетинг навчання і підтримка
- збірки кращих практик та тематичні дослідження
- доступ до фінансування
- маркетинг і реклама

Нажаль багато з зазначених недоліків знаходяться поза межами даної роботи, і як результат їм не буде приділено тої уваги на яку вони безумовно заслуговують. Але це дозволить сфокусуватися на безпосередньому вирішенні питання створення даної Web-орієнтованої системи обробки замовлень. Аналогів вищевказаним **HeavenlyPetStore, Pets Corner** та **Zoo Plus** – не має. Тому метою даної роботи є створення такого програмного продукту, покладаючись на досвід вже існуючий, та задіюючи найновітніші й ефективніші з існуючих технологій, а також уникаючи без суттєвих на те причин збільшення вартості розробки такого продукту. На меті є створення Web-орієнтованої системи обробки замовлень зоомагазину на основі freeware.

Оскільки дана Web-орієнтованої системи обробки замовлень в цілому являє собою розгалужену систему, розподілену на модулі – то в даній роботі не є можливим розглянути всі її модулі, та приділити увагу всім безпосередньо важливим її частинам. Саме тому центром уваги є – модуль бази даних. Вся інформація про товари, послуги, сервіси, користувачів та замовлення, зберігається саме в базі даних. Налаштування швидкої та надійної роботи такої системи є одним із найважливіших завдань які ставить перед собою розробник програмного продукту. А якщо взяти до уваги те що з ростом популярності навантаження у вигляді користувацьких запитів – зростатиме, то це потребуватиме оптимізованих і швидкобійних механізмів пошуку в базі даних, а також дружніх для користувача Web-інтерфейсів та інтуїтивно зрозумілих способів взаємодії із ними.

					ІАЛЦ.045490.004 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підп.	Дата		

## 2. ОПИС ІНСТРУМЕНТАРІЮ

### 2.1 Операційна система

У якості операційної системи (ОС) підходить будь-яка з нині існуючих, оскільки за мету ставилось зберегти мультиплатформенність без нанесення шкоди ефективності. Не дивлячись на те що Web-орієнтована система розроблялася використовуючи Microsoft Windows, це ніяким чином не відображається на тому, яку операційну систему може використовувати користувач аби без перешкод користуватися даним програмним продуктом. Linux чи MacOS також задовольняють усі критерії необхідні для розробки і комфортного використання. Безумовно в світі близько 70% персональних комп'ютерів використовують ту чи іншу версію MS Windows, тому доцільно розробляти програмне забезпечення опираючись на неї, але оскільки продукт, а саме Web-орієнтована система дистрибуції туристичних послуг є Web продуктом, а подальші технології також дозволяють зберегти платформну незалежність і гнучкість, то в даному випадку вибором операційної системи можна знехтувати.

### 2.2 Середовище розробки, платформа та мова програмування

Торгові системи сьогодення вимагають швидкого виконання, гнучкості та відкритості до інновацій. Саме з цієї причини використовується Java в якості реалізаційної платформи.

Написані на Java програми і додатки виконуються так само швидко, або швидше, ніж на будь-якій іншій мові.

Java надає засоби для швидкого розвитку надійних, великих і складних додатків, які легко розширити, забезпечуючи короткий час виходу на ринок початкових систем, а також дозволяє подальше системне вдосконалення протягом усього терміну служби системи. Це гарантує захист інвестицій, низькі експлуатаційні витрати і низьку сукупну вартість розробки.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		17

Java дозволяє використовувати як перевірені так і нові технології. Java дозволяє розробляти виключно швидко програмне забезпечення яке так само виключно швидко потім працює.

- Швидке виконання

Java швидко виконується, так як рідний код програми Java безперервно повторно оптимізується не тільки для виконуючого обладнання, але також і відносно програмного забезпечення, що її використовує тобто, спосіб взаємодії і розподіл ресурсів між частинами програмного забезпечення, які насправді використовуються в даний момент. Не існує іншої комерційної мови програмування, що могла б дозволити швидше. Виконання і зберегти високу оптимізацію процесів.

- Швидка розробка

Розробка на Java – це дійсно швидко, так як Java легко задовольняє будь-який стиль програмування. Це дає високу якість програмного коду при відносно коротких циклах розвитку, і дозволяє пропонувати дуже короткий розробки програмної системи загалом.

Серед плюсів також і існування великого вибору розвинутих і продуктивних інструментів високої якості, доступних для Java, розробка за допомогою яких сприяє простоті як самої розробки так і подальшому технічному обслуговуванню.

Інші мовні середовища пропонують лише частину діапазону інструментів, які доступні для Java.

- Великі системи

Java особливо підходить для великих систем з тривалим циклом життя. Java заохочує ясний стиль програмування, що робить Java-програми легкими для зберігання і підтримки при відносно низькій загальній вартості розробки.

- Гнучкість

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		19



Крім того, продовжується розвиток віртуальної машини Java (JVM) і це означає, що створені на основі Java продукти можуть часто і досить швидко скористатися новими підвищеннями продуктивності Java, при цьому нічого не змінюючи в коді, навіть без повторної компіляції, достатньо просто розгорнути їх, використовуючи новий JVM. Те ж саме стосується і нових апаратних та операційних систем архітектури. Системи створені на Java легко переміщуються зі старих архітектур в сучасні з тільки поверхневими оновленнями процесів розгортання. А розвиток Java як мови програмування надає впевненості, що вищевказане продовжуватиме проводитись в осяжному майбутньому.

- Напишіть один раз, запускайте всюди

Оскільки байт-код не залежить від фактичного цільового обладнання, віртуальна машина виконує одну з основних цілей мови Java: створення мови дійсно вільної від апаратної залежності, і програми на якій будучи одноразово написані, можуть бути запущені на будь-якому комп'ютері.

#### Допоміжні бібліотеки

*Spring Framework* - це програмний каркас (фреймворк) з відкритим кодом та контейнера з підтримкою інверсії управління для платформи Java. Основні особливості Spring Framework можуть бути використані будь-яким застосунком Java, але є розширення для створення веб-застосунків на платформі Java EE. Spring Framework складається з декількох модулів, які надають широкий спектр послуг:

- Контейнер Інверсії управління: Конфігурація компонентів додатків і управління життєвим циклом об'єктів Java, здійснюється головним чином через Інверсію управління
- Аспекто-орієнтоване програмування: дозволяє реалізувати наскрізні процедури
- Доступ до даних: робота з реляційної системи управління базами даних на платформі Java з використанням JDBC і об'єктно-реляційні відображення та інструментів з NoSQL баз даних

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		20

- Управління транзакціями: об'єднує кілька API, управління транзакціями та координує операції для Java-об'єктів
- Модель-Вигляд-Управління (Model-View-Controller): програмний каркас на основі HTTP сервлета, що забезпечує створення веб-додатків і веб-служб RESTful.
- Аутентифікація і авторизація: настроювані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик за допомогою підпроекту Spring Security.
- Віддалене управління: конфігураційний вплив і управління Java-об'єктами для місцевої (локальної) або віддаленої конфігурації через JMX
- Тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів

#### *Spring Batch*

Багато додатків вимагають об'ємної обробки для виконання бізнес-операцій в критично важливих середовищах. Ці бізнес-операції включають в себе автоматизований комплекс обробки великих обсягів інформації, що найбільш ефективно оброблені без взаємодії з користувачем. Ці операції, як правило, включають в себе (розрахунки, повідомлення або листування), періодичне застосування складних бізнес-правил обробляється повторно через дуже великий набір даних, або шляхом інтеграції інформації, отриманої від внутрішніх і зовнішніх систем, і що, як правило, вимагає форматування, перевірки та обробки транзакцій. Пакетна обробка використовується для обробки мільярдів транзакцій кожен день на будь-якому підприємстві.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		21

Spring Batch - легка, всеосяжна допоміжна бібліотека призначена для забезпечення розвитку стійких пакетних додатків, життєво важливих для щоденних операцій Веб-орієнтованих систем. Вона спирається на продуктивність, POJO на основі підходу до розробки, і загальну простоту використання можливостей для тих людей, що вже знайомі зі Spring Framework, це робить її використання легким для розробників.

Spring Batch забезпечує багаторазові функції, які необхідні при обробці великих обсягів записів, у тому числі реєстрації / відстеження, управління транзакціями, обробки статистики роботи, роботи перезапуску, пропуск і управління ресурсами. Вона також забезпечує більш технічні послуги та функції, які дозволяють надзвичайно просто досягти високої продуктивності пакетних завдань, за рахунок методів оптимізації та поділу.

Spring Batch, як правило читає велику кількість записів з бази даних, файла, або черги, обробляє дані в деякому роді, а потім записує назад дані в зміненому вигляді, як правило, в автономному середовищі без будь-якої взаємодії з користувачем. Пакетні завдання є невід'ємною частиною більшості Веб-орієнтованих систем і Spring Batch значно полегшує весь процес пакетної обробки. Одним із великих плюсів даних бібліотеки є те що вона завжди була тільки з відкритим вихідним кодом, що забезпечує надійні, корпоративного масштабу рішення.

*Spring Security* являє собою потужний механізм аутентифікації і контролю доступу. Це де-факто стандарт безпеки створених за допомогою бібліотек Spring програм.

Spring Security є допоміжною бібліотекою, яка фокусується на наданні як аутентифікації так і авторизації для програм Java. Як і всі Spring проекти, реальна сила Spring Security знаходиться в тому, як легко вона може бути розширена для задоволення вимог користувача.

Особливості:

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		22

- Всеосяжна і розширена підтримка для як аутентифікації так і авторизації
- Захист від атак типу фіксації сесії, підробки запиту, і т.д.
- Інтеграція Servlet API
- Можливість інтеграції з Spring Web MVC

*Spring Data JPA*, частина великої родини Spring Data, дозволяє легко і ефективно реалізувати репозиторії на основі JPA. Цей модуль має справу з розширеною підтримкою на основі JPA шарів доступу до даних. Це дозволяє йому легше будувати Spring-додатки, що використовують технології доступу до даних.

Реалізація програми для доступу до даних завжди була громіздкою. занадто багато стандартного коду повинно бути написано для виконання простих запитів, а також виконання розбивки на сторінки, внесення змін. Spring Data JPA покликана значно поліпшити реалізацію доступу до даних за рахунок зменшення зусилля до того рівня, що насправді необхідний. Як розробник ви пишете інтерфейси репозиторію, у тому числі користувацькі методи пошуку, і Spring буде забезпечувати їх подальшу реалізацію автоматично.

Особливості:

- Підтримка створення репозиторіїв на основі Spring та JPA
- Підтримка Querydsl предикатів і, таким чином, типізованих запитів JPA
- Підтримка прозорості об'єктної моделі
- Підтримка динамічних запитів, здатність до інтеграції користувацьких методів і процедур доступу до даних

*Hibernate* — засіб відображення між об'єктами та реляційними структурами (object-relational mapping, ORM) для платформи Java. Hibernate є вільним програмним забезпеченням, яке поширюється на умовах GNU Lesser General Public License. Hibernate надає легкий для використання каркас

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		23

(фреймворк) для відображення між об'єктно-орієнтованою моделлю даних і традиційною реляційною базою даних.

Основні можливості:

Метою Hibernate є звільнення розробника від значних типових завдань із програмування взаємодії з базою даних. Розробник може використовувати Hibernate як при розробці з нуля, так і для вже існуючої бази даних.

Hibernate піклується про зв'язок класів з таблицями бази даних (і типів даних мови програмування із типами даних SQL), і надає засоби автоматичної побудови SQL запитів й читування/запису даних, і може значно зменшити час розробки, який зазвичай витрачається на ручне написання типового SQL і JDBC коду. Hibernate генерує SQL виклики і звільняє розробника від ручної обробки результуючого набору даних, конвертації об'єктів і забезпечення сумісності із різними базами даних.

Hibernate забезпечує прозору підтримку збереження даних, тобто їхньої персистентності (англ. persistence) для «POJO»-об'єктів, себто для звичайних Java-об'єктів; співставлення, Java класів з таблицями бази даних здійснюється за допомогою конфігураційних XML файлів або Java анотацій. При використанні файлу XML, Hibernate може генерувати скелет вихідного коду для класів тривалого зберігання (persistent). У цьому немає необхідності, якщо використовується анотація. Hibernate може використовувати файл XML або анотації для підтримки схеми бази даних.

Забезпечуються можливості з організації відношення між класами «один-до-багатьох» і «багато-до-багатьох». На додаток до управління зв'язками між об'єктами, Hibernate також може керувати рефлексивними асоціаціями, де об'єкт має зв'язок «один-до-багатьох» з іншими примірниками свого власного типу даних.

Hibernate підтримує відображення користувацьких типів значень. Це робить можливим такі сценарії:

- Перевизначення типу за замовчуванням SQL, який Hibernate вибирає при відображенні стовпчика властивості.

					ІАЛП.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		24

- Картування перераховуваного типу Java до колонок БД, так ніби вони є звичайними властивостями.
- Картування однієї властивості в декілька колонок.

Hibernate забезпечує прозоре збереження POJO (Plain Old Java Objects - простих старих об'єктів Java). Єдина сувора вимога для персистентного класу — конструктор без аргументів, не обов'язково публічний. Для правильної поведінки деяких програм також потрібна особлива увага до методів equals() і hashCode().

Колекції об'єктів даних, як правило, зберігаються у вигляді колекцій Java-об'єктів, таких як набір (Set) і список (List). Підтримуються узагальнені класи (Generics), введені в Java 5. Hibernate може бути налаштований на відкладені завантаження колекцій. Відкладені завантаження є варіантом за замовчуванням, починаючи з Hibernate 3.

Зв'язані об'єкти можуть бути налаштовані на каскадні операції. Це може скоротити час розробки і забезпечити цілісність. Функція перевірки зміни даних дозволяє уникнути непотрібного запису дій в базу даних, виконуючи SQL оновлення тільки при зміні полів персистентних об'єктів.

### Front-end технологія

У якості Front-end технологій вибрано JavaScript, CSS3, HTML5.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі —

					ІАЛЦ.045490.004 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підп.	Дата		

сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

Переваги CSS:

- Інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
- Різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
- Сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;
- Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузері здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		26

роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.)

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

*JavaScript (JS)* — динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java і C#), стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		27



JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг (currying), анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має С-подібний синтаксис, але має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів
- функції як об'єкти першого класу
- обробка винятків
- автоматичне приведення типів
- автоматичне прибирання сміття
- анонімні функції

*HTML5* — наступна версія мови HTML. До складу робочої групи з HTML5 увійшли AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera та кілька сотень інших виробників.

Специфікації HTML5 не обмежуються тільки розміткою і включають в себе низку веб-технологій, котрі у сукупності формують відкриту Веб-платформу — програмне оточення для роботи крос-платформових застосунків, здатних взаємодіяти з обладнанням, і які підтримують засоби для роботи з відео, графікою і анімацією, що надає розширені мережеві можливості.

Розмічаючи області на сторінці за допомогою певних елементів, ця технологія може допомогти полегшити користувачеві навігацію. Наприклад, він може легко пропустити розділ навігації або швидко переходити від однієї статті до іншої без необхідності для авторів робити відповідні посилання. Автори також отримують вигоду в результаті заміни великої кількості div-ів одним з декількох відповідних елементів, що також приводить до чистого і легкого для автора початкового коду.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		28

Елементами header є заголовки розділів. Вони можуть складатися з декількох частин — наприклад, було б виправдано розділяти блок заголовка на підзаголовки, історію версій або вказання авторства. Елемент footer визначає нижню частину розділу, до якого він відноситься. Зазвичай він містить інформацію про розділ — наприклад, ім'я автора, посилання на схожі документи, копірайт і тому подібне. Блок nav містить список посилань для навігації. Підходить, наприклад, для навігації по сайту, або для змісту. Елемент aside підходить для розміщення вмісту яким-небудь чином спорідненого основному контенту. У звичайному випадку буде корисний для розмітки бічної колонки. Тег section представляє загальний розділ документа або додатку, наприклад, такий як розділ. Тег article відзначає незалежний розділ документа, сторінки або сайту. Застосуємо для такого вмісту як новини, запису блога, повідомлення у форумі або коментарі користувачів. Крім того, HTML5 дозволяє розробникам зберігати дані на стороні клієнта. Це значно зменшить обсяг інформації, що передається між веб браузером та сервером при використанні web-2.0 застосувань.

Вміст будь-якого елемента на сторінці можна легко поміняти за допомогою властивості contentEditable. Цей метод широко використовується при створенні WYSIWYG застосувань.

### 2.3 Система управління базами даних

Серед систем управління базами даних вибрано MySQL — вільну систему керування реляційними базами даних.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL виникла як спроба застосувати mSQL до власних розробок компанії: таблиць, для яких використовувалися ISAM — підпрограми низького рівня. У результаті був вироблений новий SQL-інтерфейс, але API-інтерфейс залишився в спадок від mSQL.

Після поглинання у 2009 році Sun Microsystems компанією Oracle Corporation MySQL стала власністю Oracle.

За час розвитку під орудою Oracle все більше відокремлює MySQL від спільноти і робить процес розробки все менш прозорим. Наприклад, повернута практика поставки власницьких розширених функцій в Enterprise-версії MySQL, спостерігається приховування інформації про вразливості, зі складу виключений тестовий набір, закритий доступ до більшої частини системи відстеження помилок та припинено публікація згрупованого логу змін, що дозволяє судити про прив'язку патчів до конкретних змін.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		30

Версія MySQL 5.1 продовжує шлях до стандарту SQL:2003. MySQL 5.1 містить такі нововведення:

- Сегментування — можливість розбити одну велику таблицю на декілька частин, розміщених в різних файлових системах, базуючись на визначеній користувачем функції. При деяких умовах це може дати серйозне збільшення продуктивності та, крім того, полегшує масштабування таблиць.
- Змінено поведінку ряду операторів для забезпечення більшої сумісності зі стандартом SQL:2003.
- Порядкова реплікація (row-based реплікація), при якій в бінарний лог буде записуватись тільки інформація про реально змінені рядки таблиці замість оригінального (і, можливо, більш повільного) тексту запиту.
- Вбудований планувальник робіт, що періодично запускаються. По синтаксису додання задачі схоже на додання тригера до таблиці; по ідеології — на crontab.
- Додатковий набір функцій для обробки XML, реалізація підтримки XPath.
- Нові засоби діагностики проблем і утиліти для аналізу продуктивності. Розширено можливості з керування вмістом лог-файлів, логи тепер можуть бути збережені і в таблицях general\_log і в slow\_log. Утиліта mysqlslap дозволяє провести тестування навантаження БД із записом часу реакції на кожний запит.
- Для спрощення операції оновлення підготовлена утиліта mysql\_upgrade, яка виконає перевірку всіх існуючих таблиць на предмет сумісності з новою версією, і при необхідності виконає належні коригування.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		31

- MySQL Cluster тепер йде як окремий продукт, який базується на MySQL 5.1 і сховищі NDBCLUSTER.
- Значні зміни в роботі MySQL Cluster, такі, як, наприклад, можливість зберігання табличних даних на диску.
- Повернення до використання вбудованої бібліотеки libmysqld, відсутньої в MySQL 5.0.
- API для плагінів, що дозволяє завантажувати сторонні модулі, які розширюють функціональність (наприклад, повнотекстовий пошук), без перезапуску сервера.
- Реалізація парсера повнотекстового пошуку у вигляді plug-in.
- Новий рушій таблиць Maria (стійкий до збоїв клон MyISAM), який у 2010 був перейменований на Aria та став основою форку MySQL від Монті Віденіуса під назвою MariaDB
- Використання за замовчуванням рушія InnoDB.
- Підтримка напівсинхронного (semi-synchronous) механізму реплікації, заснованого на патчах до InnoDB від компанії Google.
- Поліпшення функцій з партіціювання даних. Розширений синтаксис для розбиття великих таблиць на кілька частин, розміщених в різних файлових системах (partitioning). Додані операції RANGE, LIST і метод оптимізації «partition pruning».
- Новий механізм оптимізації вкладених запитів та операцій JOIN.
- Перероблена система внутрішніх блокувань.
- Інтегровані патчі Google з оптимізацією роботи InnoDB на CPU з великою кількістю ядер.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		32

### *Тип таблиць Maria*

Maria (починаючи з версії 5.2.x — Aria) — новий тип таблиць MySQL для зберігання даних. Maria являє собою розширену версію сховища MyISAM, з додаванням засобів збереження цілісності даних після краху.

Основні переваги Maria:

- У разі краху проводиться відкат результатів виконання поточної операції або повернення в стан до команди LOCK TABLES. Реалізація через ведення логів операцій.
- Можливість відновлення стану з будь-якої точки в журналі операцій, включаючи підтримку CREATE/ DROP/ RENAME/ TRUNCATE. Може бути використана для створення інкрементальних бекапів, через періодичне копіювання лог файлу.
- Підтримка всіх форматів стовпців MyISAM, розширена новим форматом «rows-in-block», що використовують сторінковий спосіб зберігання даних, при якому дані в стовпцях можуть кешуватися.
- В майбутньому буде реалізовано два режими: транзакційний і без відображення в журналі транзакцій, для не критичних даних.
- Розмір сторінки даних дорівнює 8Кб (в MyISAM 1Кб), що дозволяє досягти більш високої продуктивності для індексів за полями фіксованого розміру, але працює повільніше у разі індексування ключів змінної довжини.

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		33

### 3. ОПИС МОДУЛЯ БАЗИ ДАНИХ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ОБРОБКИ ЗАМОВЛЕНЬ ЗООМАГАЗИНУ

#### 3.1 Загальна інформація про модулі Web-орієнтованої системи обробки замовлень зоомагазину

Мета створення Web-Орієнтованої системи обробки замовлень зоомагазину полягає в забезпеченні економічних комерційних потреб українського суспільства з огляду на популярний нині європейський курс. Сама відсутність такої Web-Орієнтованої системи ставить під загрозу функціонування галузі в Україні в цілому. І хоча дана система не ставить собі за мету повністю вирішити вищевказані проблеми, проте має на меті значно спростити взаємодії між кінцевим споживачем та е-комерційною організацією, не залежно від її форми власності.

Для початку ми розглянемо всі наявні в системі модулі, коротко оглянемо їх характеристики, для отримання картини в цілому і розуміння пріоритетності поставлених завдань.

Загальна структура Web-Орієнтованої системи обробки замовлень зображена на структурній схемі та складається з ядра системи та восьми модулів, на які вона поділяється за функціональними ознаками:

1. Керівний модуль (ядро системи) - забезпечує взаємодію всіх інших частин як єдиної системи.
2. Модуль бази даних – забезпечує управління ресурсами баз даних, відповідає за інформаційне наповнення системи, та за всі процеси пов'язані зі зберіганням, створенням, редакцією та видаленням записів.
3. Модуль реєстрації користувачів – забезпечує роботу з персональними даними користувачів.
4. Модуль роботи з замовником – відповідає за формування замовлень на основі користувацьких форм, та подальші транзакції.
5. Модуль звітів – відповідає за роботу зі звітами, їх створення, автоматизацію звітного процесу.

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		34

6. Користувацький модуль – являє собою доступну для користувача основу на веб технологіях інтерфейсну форму, базується на фронт-енд технології.
7. Модуль моніторингу процесів – забезпечує моніторинг процесів, відслідковування статусів поставлених адміністратором завдань та завантаженості системи в цілому
8. Модуль безпеки – відповідає за забезпечення процесів аутентифікації та авторизації а також відповідає за безпеку системи в цілому.
9. Адміністративний модуль – забезпечує роботу адміністратора в питаннях менеджменту як системи в цілому так і окремих її частин.

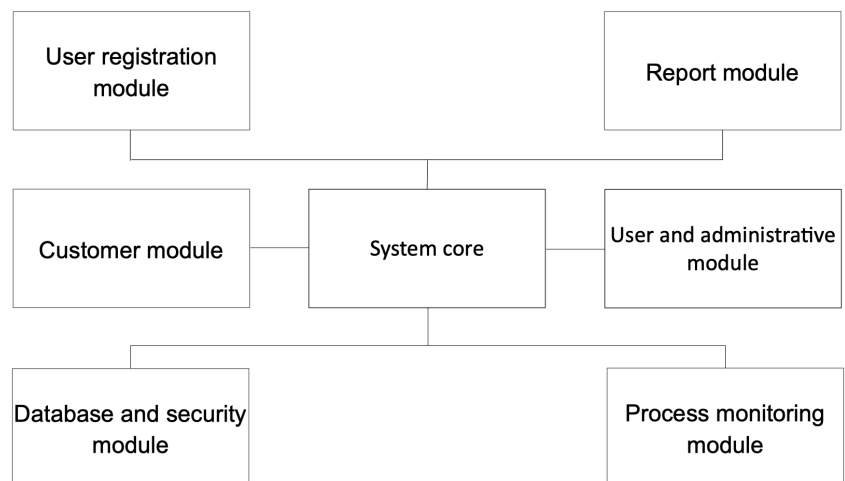


Рис.4 Структурна схема

Задачі модуля баз даних є наступними:

**Створення механізму, що дозволяє максимально автоматизувати процес обміну даними між модулями Web-Орієнтованої системи .**

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		35



Для вирішення цієї задачі використано новітні підходи до проектування баз даних, та відповідне програмне забезпечення що дозволяє одночасно мінімізувати навантаження на адмініструючий персонал, при цьому зберігаючи технологічність, гнучкість та можливість безпосередньо керувати процесами обміну даних.

**Забезпечення процесу зручного додавання, видалення, перегляду та редагування записів.**

Для вирішення цієї задачі розроблено зручний інтерфейс користувача, який без зайвих навантажень інтуїтивно дозволяє адміністратору баз даних виконувати всі необхідні операції над записами, не залежно від того чи йде мова про користувацькі персональні дані чи сформовані замовлення, чи-то списки товарів і послуг доступних на Web-Орієнтованій системі.

**Надання можливості імпортування даних про товари та послуги з CSV-файлу, з можливістю автоматичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів**

Для вирішення цієї задачі розроблено алгоритм та допоміжні сервіси та класи, які дозволяють зчитувати дані з CSV-файлу, формувати їх відповідно до стандартів записів в БД, а також при наявності зображень, що ілюструють товар чи послугу, сервіс має переходити по веб-посиланню, завантажувати автоматично вказане зображення та у разі потреби змінювати його розміри до заданих системою. Це питання розкрито в розділі 3.4.

					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		36

### 3.2 Проектування модуля бази даних.

Проектування баз даних відноситься до діяльності, яка зосереджується на розробці структури бази даних, яка буде використовуватися для зберігання і керування кінцевими користувацькими даними. База даних, яка відповідає всім вимогам користувачів не виникає просто так; її структура повинна бути ретельно розроблена. Насправді, проектування баз даних є, одним з найважливіших аспектів роботи з базами даних, Навіть хороша СУБД працюватиме не ефективно і навіть погано з недоцільно розробленою базою даних.

Дані є одним з найбільш цінних активів організації. Дані про клієнтів, співробітників, замовлення, банківські реквізити, все це життєво необхідно для існування і функціонування підприємства.

Ретельне планування має на меті забезпечити, щоб дані використовувалися належним чином і доступ до них не займав важливого часу і ресурсів.

Оскільки СУБД поточного покоління прості у використанні, то користувач отримує помилкове відчуття впевненості у своїй здатності побудувати функціональну базу даних. Ці користувачі можуть ефективно створювати об'єкти баз даних, але без правильного розуміння проектування баз даних, вони, як правило, створюють надмірно спрощені структури, які запобігають правильному зберігання даних, і як результат не відповідають реаліям, бізнесу що призводить до неповних або помилкових результатів, коли дані витягуються зі сховища.

Підхід до управління системою баз даних долає багато недоліків що уходять корінням в підходи старомодної файлової системи. Однією з ключових особливостей системи баз даних є те, що дані зберігаються в одному логічному блоці. Це означає, що, хоча дані можуть бути розподілені по кільком фізичним файлам, база даних передає ті дані що в даний час знаходяться в одному сховищі. Організація даних в одне логічне сховище дозволяє легко маніпулювати запитамі даних, на відміну від традиційних систем, де програміст повинен вказати, що саме і як вилучається зі сховища.

					ІАЛЦ.045490.004 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підп.	Дата		

При покладанні на сучасні системи баз даних, потрібно процедурно вказувати тільки те, що повинно бути зроблено, СУБД (система управління базами даних) робить все інше автоматично, без зайвого втручання оператора. Ще однією перевагою такого підходу є те, що дані знаходяться в одній базі даних, і тому зникає потреба в дублюванні. Програмне забезпечення бази даних може безперешкодно взаємодіяти з усіма даними в базі. Номери дублювання даних є одним із способів підтримки цілісності даних. Коли даним дана можливість дублювання, як правило трапляються помилки, так як у випадку коли один примірник даних змінений то як мінімум ще один примірник залишається в застарілому стані. Дублювання даних найважчим чином впливає на можливості технічного обслуговування та вимагає більше системних ресурсів для того, щоб дані перебували в синхронізованому стані.

Але все ж таки однією з найбільших переваг є те, що інформація в базах даних може бути загальною або закріпленою серед користувачів або додатків. Існує більше контролю та підзвітності над тим, як дані і де знаходяться, які операції можливо провести, оскільки згідно до логічної структури все знаходиться в одній базі даних.

Звісно є і деякі недоліки в системах баз даних, насамперед необхідність в більш потужному і складному програмному забезпеченні, що може призвести до ускладнень в керуванні базою даних і проектуванні програмного забезпечення. Також потрібні більш глибокі знання про те, як використовувати базу даних, що іноді робить систему баз даних менш зручною, ніж традиційні файлові системи. Оскільки база даних є одним логічним сховищем, навіть невелика помилка може призвести до пошкодження всієї бази даних і зменшити цілісність інформації що в ній зберігається.

Еталоном хорошої бази даних, є її повнота, простота, зрозумілість, гнучкість та функціональність. Слідуючи методології розробки програмного забезпечення життєвого циклу бази даних, і за допомогою моделі даних, проектування модулю баз даних Web-Орієнтованої системи дистрибуції туристичних послуг виконані враховуючи вищевказані недоліки що дозволить звести до мінімуму їх вплив.

					ІАЛП.045490.004 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підп.	Дата		

Етапи розробки будь-якої програми можуть бути представлені у вигляді лінійної послідовності, де кожен крок у послідовності є функцією. Прихильність до "моделі водоспаду" забезпечує якісне програмне забезпечення, яке відповідає вищевказаним критеріям (повнота, простота, зрозумілість, гнучкість та функціональність). Ці риси і є основними засадами добре побудованої бази даних. Модель водоспаду може бути застосована до теорії проектування баз даних, так само ефективно, як вона застосовується до іншої теорії розробки програмного забезпечення. Кроки можуть бути підсумовані наступним чином:

Технічне завдання -> аналіз -> Концептуальне проектування -> Реалізація Дизайну > Фізична схема Дизайну та оптимізації

У консультації з усіма потенційними користувачами бази даних, першим кроком дизайнера баз даних є розробка документа вимог даних. Документ містить короткі вимоги і являє собою технічне резюме, з зазначенням які елементи даних будуть зберігатися в базі даних, і, як різні елементи даних пов'язані один з одним. Беручи до уваги «вимоги даних», проводиться подальший аналіз, щоб, наприклад визначити більш детальні атрибути даних і при необхідності визначити обмеження. Результат цього аналізу є попереднім технічним документом. Беручи в подальшому документ специфікації, модель бази даних, надалі належить зосередитися на інформації, а саме як вона обробляється і передається для кінцевого користувача. Після проведення такого проектування, дизайнер(архітектор) переходить до вибору тих технологій що можуть задовольнити поставлені задачі.

### 3.2.1 Моделі даних і схеми в якості засобу збору даних

Проектні етапи розвитку бази даних використовують поняття "моделі даних". Моделі даних є діаграми або схеми, які використовуються, щоб представити вимоги даних на різних рівнях абстракції. Першим кроком у базі даних життєвого циклу розробки є складання документа вимог.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		39

Документ вимог як правило повинен бути проаналізований і перетворитися на основний набір даних який може бути перетворено в подальшому на концептуальну модель. Кінцевим результатом концептуальної стадії проектування є безпосередньо концептуальна модель даних, яка являє собою стислу інформацію про те, як в кінцевому підсумку буде реалізовано модуль баз даних.

На стадії проектування реалізації, концептуальна модель даних перетворюється в "логічне" уявлення бази даних. Логічна модель даних передає «логічну структуру і функціонування» бази даних і описує «як зберігаються дані (наприклад, які таблиці використовуються, які обмеження застосовуються), але не є специфічними для будь-якої СУБД. Логічна модель бази даних є концептуальна модель нижнього рівня, і яка повинна бути в подальшому покладена в основу створення фізичних конструкцій.

Фізичне моделювання має справу з «представницькими аспектами» і «оперативними аспектами» бази даних, тобто внутрішні СУБД конкретних операцій і процесів і, як СУБД взаємодіє з даними, базою даних і користувачем. Переклад з логічного проектування до фізичної конструкції призначає "функції" як для (СУБД) так і для користувача, звично це такі функції, як зберігання та безпека, а також додаткові аспекти, такі як послідовності (даних). Практично кажучи, фізична схема є наборами кодів SQL, що використовуються для створення бази даних.

Єдиним еталоном хорошої бази даних, є повнота, простота, зрозумілість, гнучкість та функціональність. Моделювання бази даних прагне до незастарілого, уніфікованого представлення всіх даних, що використовуються в організації. Слідуючи вище вказаній методиці, і за допомогою моделі даних, ці бази даних створюються згідно ідеалів дизайну.

### **Оператори SQL - Реалізація бази даних**

Останнім кроком є фізична реалізація тієї логічної конструкції, що була описана в попередньому розділі. Для фізичної реалізації бази даних, може бути використаний SQL. Основними кроками в реалізації бази даних є наступне:

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		40

### 1. Створення таблиць бази даних

Таблиці створюються безпосередньо з інформації, що міститься в словниках даних. Кожен наступний блок коду представляє собою рядок в словнику даних і виконуються один за іншим. Наприклад, блок "створити таблицю" містить детальну інформацію про всі елементи даних (фірма, товар, послуга замовлення, користувач і т.д.), їх атрибутів (імен, віку, витрат, цифр та інших деталей), відносини між елементами даних, а також ключі і правила доступу. Вся ця інформація вже описана в словнику даних, і ми за допомогою команд фактично перетворюємо його на реалізацію фізичної моделі бази даних.

```
create table PURCHASES (  
  company_name varchar2(19) not null,  
  supplier_name varchar2(19) not null,  
  purchase_date date,  
  purchase_cost number (6,2),  
  ingredient_name varchar2 (19),  
  ingredient_quantity number (3,0),  
  CONSTRAINT PURCHASES_company_name_fk FOREIGN KEY(company_name) REFERENCES  
  COMPANY(company_name),  
  CONSTRAINT PURCHASES_supplier_name_fk FOREIGN KEY(supplier_name) REFERENCES  
  SUPPLIER(supplier_name),  
  CONSTRAINT PURCHASES_company_name_pk PRIMARY KEY(company_name,  
  supplier_name)  
);
```

### 2. Заповнення таблиць

Для заповнення створених таблиць використовуються SQL команди з метою зберігання в них конкретних даних (наприклад, імен користувачів, ціни на послуги, вартість замовлення і т.д.).

### 3. Запити до бази даних.

Надалі слід написати SQL команди для отримання інформації та даних зі сховища, наприклад для побудови в подальшому каталогу туристичних послуг, чи загального списку користувачів згідно запитів адміністратора і т.д.

					ІАЛП.045490.004 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підп.	Дата		

### 3. Web-інтерфейс модуля бази даних. Керівництво користувача.

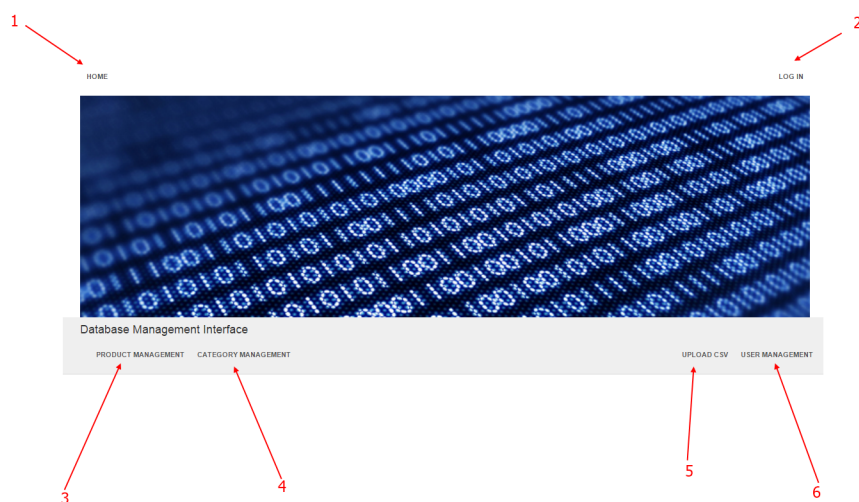


Рис. 5 Приклад роботи з Web-інтерфейсом користувача модуля бази даних

Web-інтерфейс користувача модуля бази даних, зображений на рис. 5 дозволяє адміністратору баз даних проводити подальші роботи та операції з модулем баз даних web-орієнтованої системи.

Робота розпочинається з того, що оператор баз даних або адміністратор повинні попередньо увійти до системи шляхом авторизування в ній, як ми можемо побачити на рис. 6.

#### Login with Username and Password

User:  (pointed to by red arrow 1)

Password:  (pointed to by red arrow 2)

Login (pointed to by red arrow 3)

Зм.	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045490.004 ПЗ

Арк.

42

Рис. 6 Авторизація користувача web-орієнтованої системи.

В полі User (1) – користувач повинен ввести своє користувацьке ім'я, а в полі Password (2) – відповідний пароль. Після цього логування до системи відбувається шляхом натискання кнопки Login (3).

Система перевіряє чи дані ім'я та пароль наявні в базі даних, і якщо так користувач отримує доступ до Web-інтерфейсу користувача модуля бази даних.

Як ми можемо побачити на рис. 5 Web-інтерфейс користувача модуля бази даних має такі керівні кнопки:

1. **Home** – натискання даної кнопки повертає користувача до основного екрану Web-інтерфейсу користувача модуля бази даних
2. **Login** – натискання даної кнопки дозволяє користувачу проводити авторизування до системи.
3. **Product Management** – натискання даної кнопки дозволяє адміністратору бази даних web-орієнтованої системи проводити керівні дії щодо товарів та послуг доступних для користувачів.
4. **Category Management** - натискання даної кнопки дозволяє адміністратору бази даних web-орієнтованої системи проводити керівні дії щодо категорій товарів.
5. **Upload CSV** – дана кнопка використовується коли виникає необхідність провести завантаження списків товарів, користувачів чи категорій з зовнішніх джерел, файлів, електронних таблиць.
6. **User Management** - натискання даної кнопки дозволяє адміністратору бази даних web-орієнтованої системи проводити керівні дії щодо користувачів та їх персональних даних.

**Web-інтерфейс управління продуктами.**

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		43



Щоб розпочати роботу з управління продуктами доступними для користувачів web-орієнтованої системи обробки замовлень проводиться перехід до сторінки зображено на рис.7. Побудова даного списку відбувається завдяки запиту програми до бази даних, описаних в процедурі

**ProductService:**

```
public List<Product> findAllProducts() {
    return Lists.newArrayList(productRepository.findAll());
}
```

Яка проводить запит до бази даних, та вилучає перелік всіх наявних продуктів. Список з'являється у вигляді що продемонстровано на рис. 6. В першій колонці зазначено ідентифікаційний номер товару, а в наступній – його найменування, далі розміщенні кнопки для проведення операцій над товарами.


Product management				
ID	Product Name	View	Edit	Delete
1	Grooming service	View 	Edit 	Delete 
2	Toys	View	Edit	Delete
3	Food	View	Edit	Delete
4	Vet	View	Edit	Delete
5	Colar	View	Edit	Delete

Рис. 7 Приклад роботи з Web-інтерфейсом користувача модуля бази даних, управління продуктами.

Для існуючих у списку продуктів доступні такі операції:

- Видалити продукт: кнопка «Delete» (3)
- Редагувати продукт: кнопка «Edit» (2)
- Переглянути продукт: кнопка «View» (1)

					ІАЛПЦ.045490.004 ПЗ		Арк.
Зм.	Арк.	№ докум.	Підп.	Дата			44

При натисканні кнопки «Delete» відбувається видалення товару зі списку, а також за допомогою процедури deleteProduct:

```
public void deleteProduct(Product product) {  
    productRepository.delete(product);  
}
```

Процедура генерує sql-запит та видаляє продукт із бази даних.

При натисканні кнопки «Edit» відбувається перехід до сторінки редагування продукту, відбувається запит до бази даних та згідно ідентифікаційного номеру виводиться вся інформація про продукт, як зображено на рис.8:

Name:

Description:

Price:

Duration:

Image:  

Select File +

Max File Size 15MB

Category:

Featured?

[Back to management](#)

Рис. 8 Робота з продуктами. Редагування

Даний Web-інтерфейс дозволяє вносити зміни до певного продукту:

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		45

- Змінити назву продукту
- Змінити описання продукту
- Змінити ціну продукту
- Змінити тривалість
- Змінити зображення
- Змінити категорію

Всі внесені до продукту зміни буде внесено як тільки користувач натискає кнопку – Submit. Відбувається виклик процедури updateProduct

```
public Product updateProduct(Product product) {
    return productRepository.save(product);
}
```

При натисканні кнопки «View» відбувається в свою чергу перехід до сторінки перегляду продукту, здійснюється запит до бази даних та згідно ідентифікаційного номеру виводиться вся інформація про продукт, як зображено на рис.9:

#### View product

**Name:**

Grooming service

**Description:**

Get awesome haircut for your pet

**Price:**

100 USD

**Duration:**

1h

[Edit Product](#)

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		46

Рис. 9 Робота з продуктами. Перегляд

Name – містить інформацію про найменування товару чи послуги

Description – містить детальне описання товару чи послуги

Price - містить інформацію про вартість товару чи послуги

Duration - містить інформацію про тривалість товару чи послуги

Edit product – дозволяє перейти до перехід до сторінки редагування продукту, розглянутій на рис. 8.

### 3.3.2 Web-інтерфейс управління категоріями товарів та послуг.

Category Management				
ID	Category Name	View	Edit	Delete
1	Grooming for dog	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
2	Grooming for cats	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Diet food	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Create new category</a>				

Рис. 10 Приклад роботи з Web-інтерфейсом користувача модуля бази даних, управління категоріями.

Щоб розпочати роботу з управління категоріями проводиться перехід до сторінки зображено на рис.10. Побудова даного списку відбувається завдяки запиту програми до бази даних, описаних в процедурі

CategoryService:

```

public List<Category> getAllCategories() {
    return Lists.newArrayList(categoryRepository.findAll());
}

```

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		47

Яка проводить запит до бази даних, та вилучає перелік всіх наявних категорій. Список з'являється у вигляді що продемонстровано на рис. 10. В першій колонці зазначено ідентифікаційний номер категорії, а в наступній – його найменування, далі розміщені кнопки для проведення операцій над сутностями.

Для існуючих у списку категорій доступні такі операції:

- Видалити категорію: кнопка «Delete» (3)
- Редагувати категорію: кнопка «Edit» (2)
- Переглянути категорію: кнопка «View» (1)

При натисканні кнопки «Delete» відбувається видалення товару зі списку, а також за допомогою процедури deleteCategory:

```
public void deleteCategory(Category category) {  
    categoryRepository.delete(category);  
}
```

Процедура генерує sql-запит та видаляє продукт із бази даних.

При натисканні кнопки «Edit» відбувається перехід до сторінки редагування категорії, відбувається запит до бази даних та згідно ідентифікаційного номеру виводиться вся інформація про категорію, як зображено на рис.11:

Name:

Grooming service

Description:

Get your pet groomed!

[Back to management](#)

Submit



Зм.	Арк.	№ докум.	Підп.	Дата

ІАЛЦ.045490.004 ПЗ

Арк.
48

Рис. 11 Робота з категоріями. Редагування

Даний Web-інтерфейс дозволяє вносити зміни до певної категорії:

- Змінити назву категорії
- Змінити описання категорії
- Повернутися назад до основного Web-інтерфейс управління категоріями (1)

Всі внесені до категорії зміни буде проведено як тільки користувач натискає кнопку (2) – Submit. Відбувається виклик процедури saveCategory

```
public void saveCategory(Category category) {  
    categoryRepository.save(category);  
}
```

При натисканні кнопки «View» відбувається в свою чергу перехід до сторінки перегляду категорії, здійснюється запит до бази даних та згідно ідентифікаційного номеру виводиться вся інформація про продукт, як зображено на рис.12:

**View category**

**Name:**  
Diet food

**Description:**  
Here your can find diet food for your pet

[Edit Product](#)

Рис. 12 Робота з категоріями. Перегляд

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		49

Name – містить інформацію про найменування категорії

Description – містить детальне описання категорії

Edit Category (1) – дозволяє перейти до сторінки редагування категорії, розглянутій на рис. 10.

### Web-інтерфейс управління користувачами.

User management				
ID	User Name	View	Edit	Delete
1	Ted	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
2	Tracy	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Barney	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
4	Lily	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
5	Marshall	<a href="#">View</a>	<a href="#">Edit</a>	<a href="#">Delete</a>
<a href="#">Create new user</a>				

Рис. 13 Приклад роботи з Web-інтерфейсом управління користувачами.

Щоб розпочати роботу з управління користувачами проводиться перехід до сторінки зображеної на рис.13. Побудова даного списку відбувається завдяки запиту програми до бази даних, описаних в процедурі findAllUsers:

```
public List<User> findAllUsers() {  
    return Lists.newArrayList(userRepository.findAll());  
}
```

Яка проводить запит до бази даних, та вилучає перелік всіх наявних користувачів. Список з'являється у вигляді що продемонстровано на рис. 13. В першій колонці зазначено ідентифікаційний номер користувача, а в

					ІАЛЦ.045490.004 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підп.	Дата		

наступній – його найменування, далі розміщенні кнопки для проведення операцій над сутностями.

Для існуючих у списку користувачів доступні такі операції:

- Створити нового користувача: кнопка «Register new user» (1)
- Видалити користувача: кнопка «Delete» (4)
- Редагувати користувача: кнопка «Edit» (3)
- Переглянути користувача: кнопка «View» (2)

При натисканні кнопки «Delete» відбувається видалення товару зі користувача, а також за допомогою процедури deleteUser:

```
public void deleteUser(User user) {  
    userRepository.delete(user);  
}
```

Процедура генерує sql-запит та видаляє користувача із бази даних.

При натисканні кнопки «Edit» відбувається перехід до сторінки редагування користувача, відбувається запит до бази даних та згідно ідентифікаційного номеру виводиться вся інформація про користувача, як зображено на рис. 14:

**Edit**

Login:

Password:

Email:

Phone:

First name:

Last name:

Рис. 14 Робота з користувачами. Редагування

					ІАЛПЦ.045490.004 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підп.	Дата		



Даний Web-інтерфейс дозволяє вносити зміни до персональних даних користувача:

- Змінити логін користувача
- Змінити пароль користувача
- Змінити e-mail користувача
- Змінити телефон користувача
- Змінити ім'я користувача
- Змінити прізвище користувача

Всі внесені до персональних даних користувача зміни буде проведено як тільки буде натиснуто кнопку – Submit. Відбувається виклик процедури updateUser : public User updateUser(User user) {

```
    return userRepository.save(user);
}
```

При натисканні кнопки «View» відбувається в свою чергу перехід до сторінки перегляду персональних даних користувача, здійснюється запит до бази даних та згідно ідентифікаційного номеру виводиться вся інформація про користувача, як зображено на рис.15:

View profile	
ID:	1
Name:	Ted
Login:	ted_mosby
Phone:	122323455
Email:	<a href="mailto:ted_mosby@gmail.com">ted_mosby@gmail.com</a>

Рис. 15 Робота з користувачами. Перегляд

ID – містить персональний ідентифікаційний номер користувача

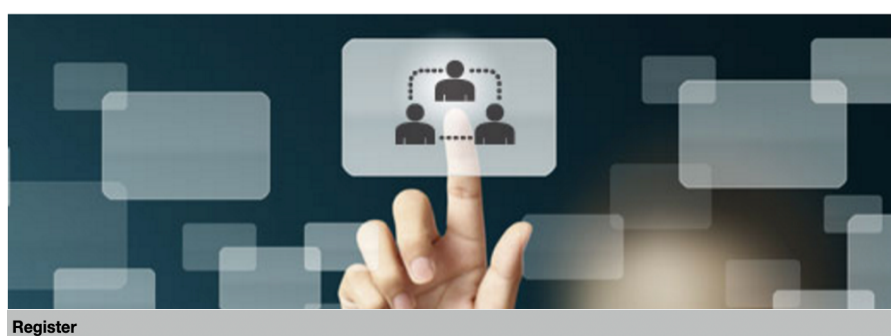
					ІАЛПЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		52

Name – містить інформацію про ім'я користувача

Login – містить інформацію про логін користувача

Phone - містить інформацію про телефон користувача

E-mail - містить інформацію про e-mail користувача



#### Register

Login:

Password:

Email:

Phone:

First name:

Last name:

Рис. 16 Робота з користувачами. Створення нового користувацького запису

Даний Web-інтерфейс дозволяє створення нового користувацького запису. Прерод тим як запис буде створено, потрібно надати наступні персональні дані:

- логін користувача
- пароль користувача
- e-mail користувача

					ІАЛЦ.045490.004 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підп.	Дата		

- телефон користувача
- ім'я користувача
- прізвище користувача

Створення нового користувацького запису буде виконано як тільки натиснуто кнопку – Submit. Відбувається виклик процедури saveUser

```
public void saveUser(User user) {
    userRepository.save(user);
    authRepository.save(new Authority(user));
}
```

#### Web-інтерфейс імпортування даних.

Upload CSV file

Job name:

Description:

Provider:

Submit

View upload reports

Рис. 17 Імпортування даних.

Щоб розпочати роботу з імпортування даних, користувачем проводиться перехід до сторінки зображеної на рис.17. Дана форма забезпечує ініціювання процесу імпорту даних, але для початку потрібно ввести наступну інформацію:

- Назву процесу : поле «Job name»
- Детальне описання процесу: поле «Description»
- Постачальник файлу з даними: поле «Provider»
- Ініціювати процес імпортування: кнопка «Submit»

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		54

Далі відбувається досить складний процес імпортування даних що буде розглянутий в подальших розділах. В цілому проводиться зчитування файлу та запис його змісту до бази даних, з можливістю розрізнення інформаційних полів та відповідного їх розміщення в іменованих колонках таблиць сховища. По закінченню процесу імпорту з'являється звіт в формі зображень на рис. 18.

Upload Status					
Job Name	Provider	Start of delivery	End of Delivery	Description	Status
1	Royal Canin	2020-10-06 15:18:36	15/6/2020 3:18:36 PM	For cats and dogs	Completed
2	Whiskas	2020-10-06 15:18:36	15/6/2020 3:18:36 PM	For cats and dogs	Completed
3	Felix	2020-10-06 15:18:36	15/6/2020 3:18:36 PM	For cats and dogs	Completed
4	Hill's	2020-10-06 15:18:36	15/6/2020 3:18:36 PM	For cats and dogs	Completed
5	Pedigree	2020-10-06 15:18:36	15/6/2020 3:18:36 PM	For cats and dogs	Completed

Рис. 18 Звіт імпортування даних.

Таблиця містить наступні поля:

- Назву процесу : колонка «Job name»
- Постачальник файлу з даними: колонка «Provider»
- Час початку процесу: колонка «Start of delivery»
- Час завершення процесу: колонка «End of delivery»
- Детальне описання процесу: колонка «Description»
- Статус процесу: колонка «Status»

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		55

#### 4. Опис алгоритмів

Основні алгоритми, що використовуються модулем баз даних, це алгоритм реєстрації користувачів у базі даних, та алгоритм імпортування даних. Більшу увагу все ж таки слід приділити першому, оскільки другий ми зможемо розглянути більш детально в розділі 3.2 де приділимо увагу питанню імпортування даних про товари та послуги з CSV-файлу, з можливістю автоматичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів.

І навпаки, детальну покрокову реалізацію алгоритму реєстрації користувачів у базі даних ми розглянемо одразу після опису основних сутностей, які приймають участь у роботі алгоритму.

1. Вхідні дані – містять наступні персональні дані: логін користувача, пароль користувача, e-mail користувача, телефон користувача, ім'я користувача, прізвище користувача.
2. Аналіз валідності даних - процес перевірки відповідності даних до стандартів системи
3. Аналіз БД на предмет унікальності користувача – перевірка наявності в БД користувачів заданим логіном
4. Створення користувача з програмним ID – безпосереднє створення класу користувача
5. Створення рівня допуску – клас що дозволяє створення буферу обміну для забезпечення більш ефективного та швидкісного запису.
6. Запис в БД – розміщення створеного класу користувача в базі даних.
7. Процес підтвердження валідності запису – перевірка успішного запису вищезазначеного класу в БД

					ІАЛЦ.045490.004 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підп.	Дата		

#### 4.1. Покрокова реалізація алгоритму

Для початку реалізації алгоритму проводиться заповнення користувачем за допомогою Веб-інтерфейсу форми, яка містить наступні персональні дані: логін користувача, пароль користувача, e-mail користувача, телефон користувача, ім'я користувача, прізвище користувача. Після того як користувач підтверджує введення даних – відбувається перевірка відповідності введених даних стандарту передбаченому класом User

```
public class User {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id", unique = true)  
    private Long id;  
  
    @NotNull  
    @Size(max = 50, min = 2)  
    private String username;  
  
    private boolean enabled;  
    private Role role;  
  
    @NotNull  
    private String password;  
  
    @NotNull  
    @Email  
    private String email;
```

Наступним кроком є запит за допомогою класу `UserController` – до бази даних та перевірка унікальності користувача.

					ІАЛІЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		57

```

@RequestMapping(method = RequestMethod.POST)
    public String saveUser(@ModelAttribute("user") @Valid Customer cus-
tomer, BindingResult result) {

        if (result.hasErrors()) {
            return "user/register";
        }
        else{
            customer.setEnabled(true);
            customer.setRole(Role.ROLE_USER);
            customerService.saveCustomer(customer);
            return "redirect:/users/" + customer.getUsername();
        }
    }

```

Як ми можемо побачити, у разі непроходження перевірки – користувач повертається до веб інтерфейсу реєстрації, а у випадку коли дані задовольняють систему – іде перехід до наступного кроку.

Наступним кроком являється створення самого класу User.

Подальший крок – створення для користувача ролі – **Authority** та прив'язки її до класу User, як ми бачимо у наступному рядку:

```
customer.setRole(Role.ROLE_USER);
```

					ІАЛІЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		58

Коли створення користувача та прав доступу для нього закінчено готовий клас передається сервісу – UserService, що виконує запис класу до бази даних, згідно з заздалегідь анотованими налаштуваннями, таким як:

```
@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", unique = true)
    private Long id;
```

Наступний крок перевірка валідності запису користувача до бази даних, у випадку якщо запис пройшов без помилок – операція створення завершується, у іншому випадку, процес починається з початку, з веб-інтерфейсу користувача.

Результатом є завершення операції, закриття всіх потоків виведення та читання, а також створення користувацького запису, що дозволяє особі-користувачу вживати свої логін та пароль для авторизації в Web-орієнтованій системі обробки замовлень та здійснювати покупки.

#### 4.2 Забезпечення імпортування даних з CSV-файлу, з можливістю авто-матичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів

Для забезпечення можливості імпортування продуктів з CSV-файлу потрібна реалізація наступних основних класів:

1. ImageDownloadService – сервіс що, використовує байтові потоки введення та виведення для проведення закачування зображення з заданого системою URL, і подальшим його розміщенням на диску.
2. BufferedImageThumbnailer - клас що відповідає за зміну розміру зображення до заданих системою.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		59



3. ImageDownloaderAndResizer – клас що об'єднує в собі обидва вищевказані, та проводить операцію автоматичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів

На початку процесу користувачем оператором вводиться шлях до CSV-файлу. Далі система аналізує дані, і перевіряє чи заданий файл існує. У випадку якщо шлях до файлу є хибним – система видає інформацію про помилку, записує її в журнал та припиняє процес імпорту, якщо ж файл існує то проводиться зчитування його першого рядку. Рядок містить інформацію яка надалі вноситься до полей класу, згідно з форматом класу. І от коли доходить черга до заповнення поля – шлях до зображення (imagePath) відбувається створення та запуск основного класу проводить операцію автоматичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів.

Шлях ImageDownloadService отримує з CSV-файлу – далі відбувається завантаження зображення за допомогою потоку вводу в тимчасовий файл, а потім за допомогою виводу – створюється зображення точно такого розміру і розподільної здатності як те що знаходиться по вказаній URL-адресі.

Саме тоді стає у нагоді BufferedImageThumbnailer - клас що відповідає за зміну розміру зображення до заданих системою. Отримуючи від ImageDownloaderAndResizer – шлях до файлу, BufferedImageThumbnailer зчитує його за допомогою байтового потоку, а потім за допомогою графічної бібліотеки проводить його зменшення до розмірів вказаних системою, фізичних для подальшого розміщення на веб-сторінці. По закінченню операції зміни розміру зображення BufferedImageThumbnailer – видає системі шлях до відформатованого файлу.

Система отримує шлях до файлу та вписує його до поля класу продукту – imagePath. Після чого клас може бути розміщений для зберігання у базі даних

					ІАЛП.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		60

Описаний механізм роботи дозволяє проводити імпортування з Csv-файлу не тільки інформаційні поля товарів, такі як наприклад назва та описання, але і забезпечувати клас зображеннями відформатованими згідно з потребами системи, при цьому операція виконується автоматично, без жодного втручання оператора.

					ІАЛПЦ.045490.004 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підп.	Дата		

## ВИСНОВКИ

Метою дипломного проекту була розробка програмного модулю управління базою даних Web-орієнтованої системи обробки замовлень зоомагазину.

В ході виконання дипломного проекту було:

- Виконано огляд існуючих рішень серед Web-орієнтованих систем обробки замовлень зоомагазинів;
- проведено аналіз проектування баз даних Web-орієнтованих систем;
- розроблено програмний модуль, який забезпечує імпортування даних про товари та послуги з CSV-файлу, з можливістю автоматичного отримання зображень з WEB-ресурсу, та подальшого їх форматування до заданих розмірів;
- запропоновано та реалізовано алгоритм реєстрації користувачів в базі даних Web-орієнтованої системи обробки замовлень зоомагазину
- запропоновано та реалізовано алгоритм імпортування даних в БД

Розроблений модуль дозволяє реєструвати користувачів, товари та послуги в базі даних, здійснювати пошук серед наявних у базі даних інформаційних блоків, виходячи з ресурсних потреб та з огляду на кількість користувацьких запитів, а також надає можливість імпортування даних до бази даних з зовнішніх ресурсів, таких як файли та інформаційні таблиці.

Програмний модуль планується до впровадження на підприємстві, що підтверджується довідкою про впровадження.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		62

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Web-програмування [Електронний ресурс]. - Режим доступу: <https://spring.io/docs/reference>
2. Database Systems: Design, Implementation, and Management; Carlos Coronel, Steven Morris, Publisher: Course Technology; 10 edition (January 1, 2012)
3. Thinking in Java, Bruce Eckel. «Prentice Hall», 2006
4. Системы баз данных. Теория и практика использования в Internet и среде Java. Грег Риккарди – изд. Вильямс, 2001 г.
5. “SQL Database Programming” (2015 Edition) By Chris Fehily
6. Head First Data Analysis: A learner's guide to big numbers, statistics, and good decisions; Michael Milton - Publisher: O'Reilly Media; 1 edition (August 7, 2009)
7. MySQL Cookbook; Paul DuBois - Publisher: O'Reilly Media; Second edition (January 27, 2007)
8. Effective Java; Joshua Bloch - Publisher: Addison-Wesley; 2 edition (May 28, 2008)
9. Горшенина Е.В., Сорокин П.А. Информационные технологии в электронной коммерции. Оргтехника. – М.: Изд-во. центр "Академия", 2002. – 240 с.
10. Pets Corner [Електронний ресурс]. - Режим доступу: <https://www.petscorner.co.uk/>
11. ZooPlus [Електронний ресурс]. - Режим доступу: <https://www.zooplus.com/>
12. HeavenlyPetStore [Електронний ресурс]. - Режим доступу: <https://heavenlypetstore.com/>

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		63

13. E-Commerce 2015 (11th Edition); Kenneth C. Laudon, Carol Traver - Publisher: Prentice Hall; 11 edition (December 5, 2014)
14. HTML and CSS: Design and Build Websites; Jon Duckett - Publisher: John Wiley & Sons; 1st edition (November 8, 2011)
15. Мови HTML і CSS для створення Web-сайтів, Моллі Є., Видавництво: Тріумф, 2007.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		64